Contract CO1: addFilePath
Operation: addFilePath(filePath: String)
Cross References : Use Case Process Trips

Pre-conditions:

- The file path is valid and points to a correctly formatted CSV file
- A CityCatalog object is created
- A ConnectionCatalog object is created

Post-conditions:

- Connection instances are created and their attributes are modified. (instance creation and attribute modification)
- City instances are created and their attributes are modified. (instance creation and attribute modification)
- Connection instances are associated with ConnectionCatalog. (formation of association)
- City instances are associated with CityCatalog. (formation of association)
- Connection instances are associated with Cities. (formation of association)

Contract CO2: editDepartureCity
Operation: editDepartureCity()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated.
- The City with the name cityName exists and has been instantiated

Post-conditions:

- The departureCity attribute of the search instance is modified (attribute modification)

Contract CO3: editArrivalCity
Operation: editArrivalCity()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated
- The City with the name cityName exists and has been instantiated

Post-conditions:

- The arrivalCity attribute of the search instance is modified (attribute modification)

Contract CO4: editDepartureTime
Operation: editDepartureTime()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated
- The day represents a valid day (Mon, Tue…)
- The time represents a valid time (HH:MM)

Post-conditions:

- The departureTime attribute of the search instance is modified (attribute modification)
- The departureDay attribute of the search instance is modified (attribute modification)

Contract CO5: editArrivalTime
Operation: editArrivalTime()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated
- The day represents a valid day (Mon, Tue…)
- The time represents a valid time (HH:MM)

Post-conditions:

- The arrivalTime attribute of the search instance is modified (attribute modification)
- The arrivalDay attribute of the search instance is modified (attribute modification)

Contract CO6: editTrainType
Operation: editTrainType()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated.
- The name represents a valid train type (TGV, ICE…)

Post-conditions:

- The TrainType attribute of the search instance is modified (attribute modification)

Contract CO7: editDaysOfOperation
Operation: editDaysOfOperation()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated.

Post-conditions:

- The daysOfOperation attribute of the search instance is modified (attribute modification)

Contract CO7: editSeatingClass()

Operation: editSeatingClass()
Cross References : Use Case Filtering Trips

Pre-conditions:

- A search object has been instantiated

Post-conditions:

- The SeatingClass attribute of the search instance is modified (attribute modification)

Contract CO8: editPriceOptions()
Operation: Use Case Filtering Trips
Cross References : Use Case Process Cost range Second Class

Pre-conditions:

- A search object has been instantiated

Post-conditions:

- The minCost and maxCost attribute of the search instance are modified (attribute modification)

Contract CO9: runSearch
Operation: runSearch()
Cross References : Use Case Search Trips

Pre-conditions:

- All the connections have been parsed by the system and are loaded
- The user has modified the search parameters to their according to their needs

Post-conditions:

- Trip objects corresponding to the Search objects' parameters are instantiated and their attributes are modified. (instance creation and attribute modification)
- The search instance is associated through a many-to-many connection to the trip instances. (association formed)

Contract CO10: editSortingOptions()
Operation: editSortingOptions()
Cross References : Use Case Sort Trips

Pre-conditions:

- A search object has been instantiated

Post-conditions:

- setSortBy and setOrder attributes of the search instance are modified (attribute modification)

Contract CO11: createClient()
Operation: createClient(firstName,lastName,age, ID)
Cross References : Use Case Book a Trip

Pre-conditions:

- A client instance for the specific client (defined by the ID) does not exist

Post-conditions:

- A client object has been created and instantiated with its attributes.
- This client object is added to the client collection inside the ClientCatalog inside.
- The client object is aggregated to the ClientCatalog instance,

Contract CO12: getTrip()
Operation: getTrip(ID)
Cross References : Use Case Book a Trip

Pre-conditions:

- A trip instance either exists.

Post-conditions:

- Return the trip instance if found otherwise return null.

Contract CO13: getClient()
Operation: getClient(ID)
Cross References : Use Case Book a Trip

Pre-conditions:

- A client instance either exists or not inside the collection of the ClientCatalog

Post-conditions:

- Return the client instance if found or return null if such a client instance does not exist.

Contract CO14: reserveTrip()
Operation: reserveTrip(client,trip)
Cross References : Use Case Book a Trip

Pre-conditions:

- A client object and a trip object exists.

Post-conditions:

- An instance of the ticket class is created all its attributes are instantiated.
- The new instance is added to the Ticket collection residing inside the TicketCatalog class.
- This instance is associated with the client and trip instance as well as the TicketCatalog.

Contract CO15: viewTrips()
Operation: viewTrips(lastName, clientID)
Cross References : Use Case View Trips

Pre-conditions:

- A client object with the lastName and clientID exists.

Post-conditions:

- The trips associated with the client are returned.